ILR #8: Progress Review 9

Zihao (Theo) Zhang- Team A

March 2nd, 2017

Teammates: Amit Agarwal, Harry Golash, Yihao Qian, Menghan Zhang

Individual Progress

Earlier this semester, our team decided to add a GPS device to our perception system in order to help with object tracking and motion estimation of the testing vehicle.

We chose the Ultimate GPS module from Adafruit, which has a sensitivity of -165 dBm and 66 channels. It can provide updates at a rate of up to 10Hz, which meets the basic requirement for real-time application. Since it was delivered last week, I have spent most of the past week on working on the GPS.

The GPS module is designed for convenient use with Arduino. I first finished all the necessary setup on hardware by properly connecting the GPS module with computer through Arduino. For the initial test, the GPS module was directly connected to the computer via the TTL serial to USB converter on the Arduino. The unparsed NMEA data was successful acquired using the Arduino Serial Monitor.

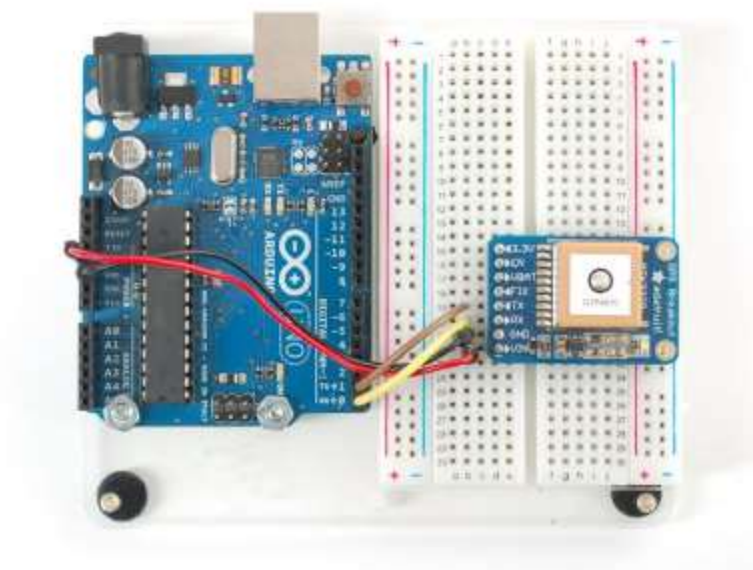

Figure 1. Hardware setup

Then, the GPS module was re-wired to the microcontroller (Arduino). Thanks to the provided library, I was able to receive parsed GPS data from the Arduino Serial Monitor by running the example program contained in the library. The data contains the current time, location (in longitude and latitude), ground speed, track angle, and altitude, all in proper formats.

As we might need to integrate the GPS data with other data from the perception system in the native Linux environment, depending on Arduino IDE/ Serial Monitor to acquire data from GPS is not a permanent solution. Therefore, I decided to start develop a software program that can acquire the data in Linux without the help of Arduino IDE. I used C/C++ to develop the program, which can fetch the serial data at the identified port and parse the data into proper format.

The raw GPS data contains NMEA sentences expressed in different formats. The two most commonly used formats are GPGGA (Global Positioning System Fix data) and GPRMC (Recommended minimum specific GPS/Transit data). I chose to use the GPRMC format because it directly provides the ground speed information, which is the most valuable for our application. The additional information provided by GPGGA, such as altitude or fix quality, does not seem to be very useful in this case.

Figure 2. Raw output data from GPS in Arduino Serial Monitor

I also took the GPS for an on-road testing, where the output speed from GPS was roughly compared with the reading from the vehicle's speedometer. Since the unit of the output speed from GPS was in knots, I finally converted it into mile-per-hour in the program.

Currently, the GPS data can be successfully acquired in the native Linux environment and is ready to be integrated with other part of the system. The update rate is set to be 1 Hz for now. However, this can be modified (< 10 Hz) based on the actual needs, with just minor changes in the program.

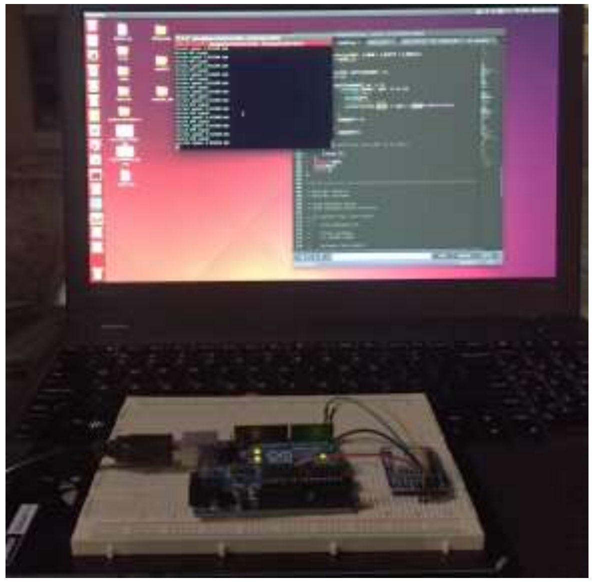

Figure 3. Outdoor Testing of the GPS module

Challenges

For the project in general, radar still remains as the biggest challenge. The progress on acquiring useful data from the radar should definitely be accelerated, as this process has been delayed so much. At the same time, the rest of the team are trying the best to work on other subsystems and modules in a way such that the progress on other parts of the system can be made steadily without much dependence on the radar.

For the power distribution PCB, the main component (LM2587 Flyback regulator) was damaged by an accident last week, so the radar can only be tested using the bulky DC generator. I have ordered three more LM 2587 regulator and an off-the-shelf DC-DC converter using the same regulator as the backup.

For the tasks on GPS module, system-level programming in C/C++ was the most challenging part, which took me some extra efforts. The tasks were achieved eventually on time with hours of learning and work, as well as some kind help from friends from other teams.

Teamwork

Amit Agarwal:

Amit has been working with Harry on the radar. They showed the ability to acquire raw (hexadecimal) data from the radar through Ethernet connection.

Harry Golash:

Harry has been working with Amit on the radar. They showed the ability to acquire raw (hexadecimal) data from the radar through Ethernet connection.

Yihao Qian:

Yihao tested a new stereo vison matching algorithm called ELAS (Efficient Large-scale Stereo). We redid the stereo vision calibration and collected new sample images with a shorter baseline. The performance of the new algorithm with the shorter baseline was compared qualitatively with the SGBM algorithm with longer baseline.

Menghan Zhang:

Menghan finished building a basic GUI for visualizing results from the perception system. The GUI can display the location and velocity of each object relative to the testing vehicle, as well as the category of the object (vehicle or pedestrian) from the top view.

Plans

Before the next ILR/ progress review, our team would like to achieve the following goals on our project of developing the perception system using Stereo Vision and radar:

1. Parsed and filtered data from the radar
2. Basic tracking for multiple objects using vision
3. Verify accuracy of GPS
4. Set up detailed plan for system integration
5. System integration in progress

In term of individual work, I will be fully responsible for verifying the accuracy of GPS (Task 3). In addition, I might help with the object tracking (Task 2), depending on the actual progress on this task. I will start to work on the system integration once after the detailed plan is set up and the integration platform is selected definitely.